

Listing of Claims:

1.-7. (cancelled)

8. (previously presented) A computer-based system for structuring, storing and processing of computer-readable data from a plurality of distinct software applications, said computer-readable data comprising hierarchically structured data set objects stored in at least one object database, said computer-readable data subject to one or more incompatible data exchange structures in the plurality of distinct software applications, said computer-readable data to be exchanged between the plurality of distinct software applications in accordance with a generic object model, wherein the object model has at least one first element which corresponds to a type Object, said computer-based system comprising:

a processor adapted to process the type Object to produce the following attributes in the type Object:

a unique identification of an object of the type Object for absolute referencing of the object,

a logical name for labeling the object, and

at least one link to a second element, which corresponds to a type Feature,

wherein the processor is further adapted to process the type Feature to produce the following attributes in the type Feature:

a unique name in relation to the object, and

a linkage to at least one of the following: further components of the type Object, further components of the type Feature, and to computer-readable data from the plurality of distinct software applications, wherein computer-readable data modeled in accordance with said generic object model comprises at least one uniformly understood object with respect to the plurality of distinct software applications, said at least one uniformly understood object being free of the one or more incompatible data exchange structures in the plurality of distinct software

applications to perform the exchange of computer-readable data between the plurality of distinct software applications without translating said computer-readable data.

9. (previously presented) The system in accordance with Claim 8, wherein the type Object has as further attributes an identification of the object type and an identification of the version of the object.

10. (previously presented) The system in accordance with Claim 8, wherein elements linked by an element of type Feature form a logical subset of all elements of an object.

11. (previously presented) The system in accordance with Claim 9, wherein elements linked by an element of type Feature form a logical subset of all elements of an object.

12. (previously presented) The system in accordance with Claim 8, wherein the elements of the object are linked by references.

13. (previously presented) The system in accordance with Claim 9, wherein the elements of the object are linked by references.

14. (previously presented) The system in accordance with Claim 10, wherein the elements of the object are linked by references.

15. (previously presented) The system in accordance with Claim 8, wherein the object model is described by an extensible markup language.

16. (previously presented) The system in accordance with Claim 15, wherein the object model is described by an extensible markup language.

17. (previously presented) The system in accordance with Claim 9, wherein the object model is described by an extensible markup language.

18. (previously presented) The system in accordance with Claim 10, wherein the object model is described by an extensible markup language.

19. (previously presented) The system in accordance with Claim 12, wherein the object model is described by an extensible markup language.

20. (previously presented) The system in accordance with Claim 8, wherein the system is part of an engineering system of an automation system.

21. (previously presented) A computer-based method for structuring, storing and processing computer-readable data from a plurality of distinct software applications, said computer-readable data comprising hierarchically structured data set objects stored in at least one object database, said computer-readable data subject to one or more incompatible data exchange structures in the plurality of distinct software applications, said computer-readable data to be exchanged between the plurality of distinct software applications in accordance with a generic object model, wherein the object model has at least one first element corresponding to the type Object, wherein the method comprises processing the type Object, said processing adapted to produce the following attributes in the type Object:

- a unique identification of an object of the type Object for absolute referencing of the object,

- a logical name for labeling the object, and

- at least one link to a second element, which corresponds to a type Feature, the method further comprising:

- assigning a unique identification to an instance of the type Object for absolute referencing the instance;

- assigning a logical name for labeling the instance; and

- linking the instance to the second element, wherein the method further comprises processing the type Feature to produce the following attributes in the type Feature:

- a unique name in relation to the relevant linked object referenced, and

- a linkage to at least one of the following: further components of the type Object, further components of the type Feature, and to computer-readable data from the plurality of distinct software applications, wherein computer-readable data modeled in accordance with said generic object model comprises at least one uniformly understood object with respect to the plurality of distinct software applications, said at least one uniformly understood object being free of the one or more incompatible data exchange structures in the plurality of distinct software applications to perform the exchange of computer-readable data between the plurality of distinct software applications without translating said computer-readable data.

22. (previously presented) The method in accordance with claim 21, wherein the data are structured, stored, and processed for engineering an automation system.

23. (previously presented) A computer-based method for structuring, storing and processing of computer-readable data from a plurality of distinct software applications, said computer-readable data comprising hierarchically structured data set objects stored in at least one object database, said computer-readable data subject to one or more incompatible data exchange structures in the plurality of distinct software applications, said computer-readable data to be exchanged between the plurality of distinct software applications in accordance with a generic object model, wherein the object model has at least one first element which corresponds to the type Object, the method comprising:

- providing a unique identification of an object of the type Object for absolute referencing of the object;

- providing a logical name for labeling the object; and

- linking the object to a second element, which corresponds to a type Feature, wherein the method further comprises processing the type Feature to produce the following attributes in the type Feature:

- a unique name in relation to the linked object, and

- a linkage to at least one of the following: further components of type Object, further components of type Feature and to computer-readable data from the plurality of distinct software applications, wherein computer-readable data modeled in accordance with said generic object model comprises at least one uniformly understood object with respect to the plurality of distinct software applications, said at least one uniformly understood object being free of the one or more incompatible data exchange structures in the plurality of distinct software applications to perform the exchange of computer-readable data between the plurality of distinct software applications without translating said computer-readable data.

24. (previously presented) The method in accordance with claim 23, wherein the data are structured, stored, and processed for engineering an automation system.